

Adaptive turning control for an agricultural robot tractor

Hao Wang¹, Noboru Noguchi^{2*}

(1. Graduate School of Agriculture, Hokkaido University, Kita 9 Nishi 9, Sapporo, Hokkaido, 065-8589, Japan;

2. Research Faculty of Agriculture, Hokkaido University, Kita 9 Nishi 9, Sapporo, Hokkaido, 065-8589, Japan)

Abstract: An adaptive turning algorithm for a four-wheel robot tractor in the headland is presented in this paper. The navigation sensors consisted of an inertial measurement unit and a real-time kinematic global positioning system (GPS). An objective function based on weights was used to create the navigation path, connecting by continuous primitives. The asymmetric steering mechanism was then taken into consideration with a vehicle model. To follow the path accurately, the slide movement of the robot and the steering rate were taken into account by estimating the turning radius in real time. In addition, the vehicle model was tuned based on the results of each turn. Therefore, the turning control algorithm was optimized on the basis of the specific conditions in the field. Field experiments showed that the robot tractor approached the next path with an average lateral deviation of 3.9 cm at a speed of 1.2 m/s during a turn. Compared to a conventional turning scheme, the time consumption and turning trajectory were decreased by 17% and 21%, respectively.

Keywords: autonomous tractor, path planning, dynamic circle-back turning, switch-back turning, robot tractor, reinforcement learning

DOI: 10.25165/j.ijabe.20181106.3605

Citation: Wang H, Noguchi N. Adaptive turning control for an agricultural robot tractor. *Int J Agric & Biol Eng*, 2018; 11(6): 113–119.

1 Introduction

Automation of agricultural machinery is considered to be one of the most efficient ways of improving productivity and quality of farming. The turning maneuver with extensive driving operations plays a crucial role in the automation of agricultural robots. Challenges of autonomous turning control for agricultural robots are to design feasible navigation path and to implement adaptive control techniques. In pursuit of stable control performance, the control algorithms should be adaptive based on the change of vehicle speed and soil conditions.

Optimal methods for headland turns include u-turning^[1], keyhole turning/bulb turn/forward turning^[2], fishtail turn/switch-back turning^[3,4] and other methods modified from previous ones according to different shapes of headland^[5-7]. The turning methods mentioned above can be easily embedded in autonomous navigation systems because maneuvers are all fully specialized. However, those methods are not adaptive for changes in various conditions such as the vehicle speed and the soil property. In practice, additional paths are added at the start or the end of turning procedures in order to smooth the planned path and to decrease the lateral deviation after turning. For example, an additional straight forward path^[4,8] and a continuous curvature path^[9-11] are added before and after headland turns. With these modifications, a robot tractor turns to the next path with a longer trajectory and occupies more headland space. These methods also require the control unit of the robot to be equipped with sufficient memory and computing capacity for designing and optimizing the

path when in operation.

Model-based control methods for wheel-type tractors are developed to tackle the straight and curved paths tracking problem^[12-15]. Recent publications also address the trajectory tracking problem of tractor-implementation systems on the basis of the kinematic model, the dynamic model, and their modifications^[13]. Bicycle model and adaptive model are applied to control the automated car performing functions, including lane keeping, lane change and overtaking^[16-18]. The vehicle motion model was used to control the automobile during the reverse movement^[19]. However, few of them have considered the sideslip of the vehicles, asymmetric steering mechanism in the left/right side or different turning radii of backward and forward movements. For example, a nonlinear model predictive control^[13] was able to estimate the slip parameters of tire-soil interactions in different environmental conditions. However, the performance was not satisfactory. An average deviation of 0.60 m and a maximum deviation of 1.54 m were found during the headland turns. Considering the changes of soil moisture, steering angle, and speed, the turning control should be optimized by adapting the control parameters on the basis of a vehicle model tuned by measured data.

To simplify maneuvers at the headland and to increase the accuracy after turning, an adaptive control algorithm called dynamic circle-back (DCB) turning is introduced in this paper. The designed operations are easy for the robot to perform. Also, considering the lock-to-lock time (LTL, the time taken to fully steer from one side to the other side) and the slide of the vehicle, the control algorithm records the position of the vehicle and calculates the turning radius in real time. By adjusting the weights of demands (high accuracy after turning, less time consumption, and less headland occupancy), the algorithm creates an optimal path according to the estimation of the forward turning radius. In addition, a reinforcement learning method is used to update the steering model after each turn. The performance of the algorithm and improvements in the algorithm compared with conventional schemes were evaluated in dynamic field tests.

Received date: 2017-06-29 **Accepted date:** 2017-12-01

Biographies: Hao Wang, PhD candidate, research interests: vehicle robotics, Email: wanghao.bit@gmail.com.

***Corresponding author:** Noboru Noguchi, PhD, Professor, research interests: vehicle robotics. Division of Fundamental AgriScience Research Faculty of Agriculture, Hokkaido University, Japan. Tel: +81-11-706-3847, Email: noguchi@cen.agr.hokudai.ac.jp.

2 Specification of adaptive turning control algorithm

The DCB turning control algorithm plans the turning path by imitating the farmer’s maneuvers. Figure 1 shows a typical trajectory created by the DCB turning. In this example, the path space, the distance between the current path (Path_1) and the next path (Path_2), is W . This pattern of the trajectory is especially suitable for a narrow path space (less than 2 times the minimum turning radius) condition that needs complex lane-change maneuvers. At the end of the working path, the tractor turns left and moves along arch A-B. Then the tractor steers to the right at point B and goes backward to point C. Finally, the tractor goes forward and approaches the next path from point D. Unlike the algorithms mentioned before, the turning radii of forward movement (R_{left}) and backward movement (R_{right}) are of different values. Furthermore, all of the turning radii are measured during turning, rather than adopting empirical values. According to the measured turning radius, the backward navigation path is optimized in real time.

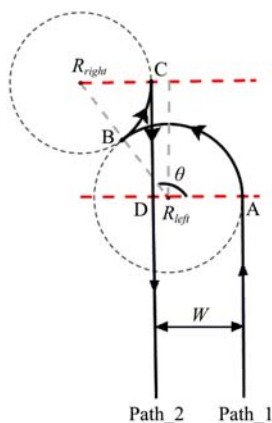


Figure 1 Scheme of the DCB algorithm

A flowchart of turning algorithm is shown in Figure 2. The input of the turning algorithm includes the position of the vehicle, the posture of the vehicle (yaw, pitch and roll angles), the speed, the steering angle, the lateral and the heading deviations from the navigation map, and limitations of speed, steering angle and headland occupancy. By balancing designer’s demands (higher accuracy, less time consumption, and smaller headland occupancy), the objective function of the dynamic turning method creates an optimal pathway according to the vehicle model. The vehicle model indicates relations of the steering angle and the turning radius in different sides with forward or backward movement. The regression analysis is used to estimate the turning radius of forward movement in real time. In addition, considering the movement during LTL time, delay response caused by the inertia of the vehicle, and the sideslip while turning, the objective function calculates the turning radius of forward movement and optimizes the pathway using the vehicle model in real time. The error of the vehicle model is represented as the difference between estimated turning radius (from the vehicle model) and the calculated turning radius (from the regression analysis). The performance of turning control is evaluated by the error analysis mechanism, which considers the error of the model and the lateral deviation at the end of backward movement (the Point C in Figure 1). After each headland turn, a reinforcement learning method is adopted in real time to tune the vehicle model by considering measured data as the training dataset.

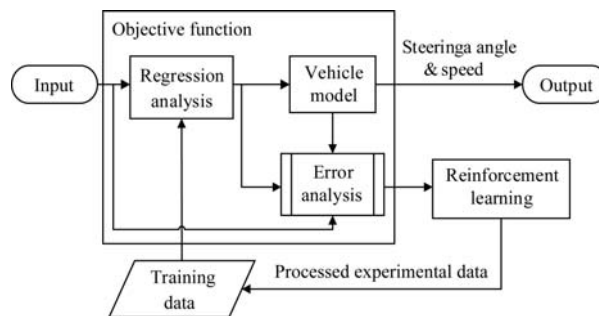


Figure 2 Flowchart of the DCB turning algorithm

2.1 Objective function based on weights

To adjust the turning route according to different regulations, such as efficiency, speed, headland space and so on, an objective function (J) is defined to optimize the time consumption (T) and the lateral deviation from navigation path (Δd) in:

$$J(T, \Delta d) = h \cdot H_0 + t \cdot T + \varepsilon \cdot \Delta d \tag{1}$$

$$s.t. \quad h + t + \varepsilon = 1$$

which takes the weights of headland distance (h), time (t), end-point accuracy (ε) requirements, and reference headland distance H_0 as inputs, and then designs the navigation path and decides the steering angle (δ) and turning speed (v). During the turning (with forward and backward movement), the vehicle keeps a constant speed determined by:

$$v = V_{min} + t(V_{max} - V_{min}) \tag{2}$$

where, V_{min} and V_{max} are the minimum and maximum speed limitations of turning. The values of weights are three hyper-parameters that need the user to adjust according to different requirements. Figure 3 shows three typical examples of paths created by different combinations of weights in the objective function. If the headland distance is short, we have to compromise on end-point accuracy and time consumption. In this case, the trajectory A-B₁-C₁-D might be an optimal choice. If time is limited, the tractor should follow a path such as A-B₃-C₃-D with less trajectory and high turning speed. Under an ideal condition, point C₂ will be on the extension line of the next path, so the lateral deviation of the vehicle would be zero along the path of C₂-D.

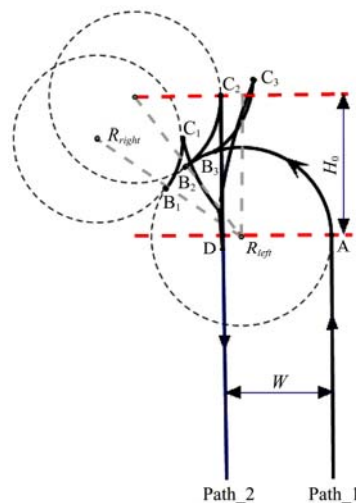


Figure 3 Planned paths for turning

Considering vertical and horizontal disturbances applied to the soil at the headland, the robot tractor starts to turn at the end of the straight path without pause. The tractor starts to steer at the end of the current path and to go forward until the yaw angle has changed θ degree:

$$\theta = 90 + \arcsin((R_b + W - R_f)/(R_f + R_b)) \cdot 180\pi + 10.0(-t + h) \cdot (1 - \varepsilon) + \theta_{bias} \quad (3)$$

where, R_f and R_b are the turning radii of forward movement and backward movement, respectively. W is the path space. Ideally, the robot will follow the path A-B₂-C₂-D (in Figure 3) when t equals h or ε equals 1. Considering the move during LTL time and the wheel slip, the control algorithm records the position of the vehicle during turning and calculates the turning radius by the least square method (LSM) in:

$$agr \min(a, b, R) = \sum_{i=1}^N (x_i - a)^2 + (y_i - b)^2 - R^2 \quad (4)$$

where, (x_i, y_i) , $i \in [1, N]$ are the 2D position of the robot tractor recorded by the real-time kinematic-global positioning system (RTK-GPS), and a , b and R are the center position of the fitted circle and the estimated turning radius, respectively. Besides the movement during LTL time and wheel slip, the vehicle cannot stop immediately at the desired position because of delay response of the control unit and the inertia of the vehicle. Therefore, the deviation after backward turning (for example, the lateral deviation^[20] at point C₂ in Figure 3) is taken as a feedback to resolve the bias of the model, as shown in:

$$\theta_{bias} = \arcsin[\Delta d / (R_f + R_b)] \quad (5)$$

where, Δd is the lateral deviation from the navigation path, and the sign of the deviation is determined by the relative position of the robot and the next working path^[4].

2.2 Learning function based vehicle model

Different from the bicycle model, in which the lateral forces in left and right wheels are simply assumed to be equal in value and direction, the vehicle model of DCB turning method deals with the movements of the tractor in different directions separately. As the vehicle reverses during turning, the actual turning radius varies according to the lateral tire force of front and rear wheels and the slip angle of the tires. Due to the asymmetric mechanism and the variance of hydraulic steering power, the turning radii of steering to the right and steering to the left are different. Additionally, the soil type or moisture, steering angle and speed affect the turning radii. Regardless of estimating unknown parameters separately, the DCB turning algorithm treats the vehicle (with or without a farming implement) as a unit, analyzes the vehicle-soil interactions through variations of turning radii and tunes the model's parameters in real time.

Table 1 shows a sample of the dataset for modeling the vehicle. The data are classified by the steering direction, the steering angle, the direction of the movement, the turning radius of the trajectory, and weights of data in each row. The values of radii in the fifth column are mean values of several experiments at different turning speeds or soil conditions. The weight of each group of data in a row is related to the accuracy of the data. The initial dataset shares the same weight, $1/N_0$ (N_0 being the total number of initial dataset). The training data plotted in Figure 4 indicates the relation of turning radii in different directions (forward and backward) at several steering angles (Figures 4a and 4b) and the relation of steering angle and turning radius (Figures 4c and 4d).

Table 1 Sample of training data

Weight	Steer	Angle/(°)	Direction	Radius/m
$1/N_0$	Left	30	Forward	5.6
$1/N_0$	Right	40	Backward	4.2
...

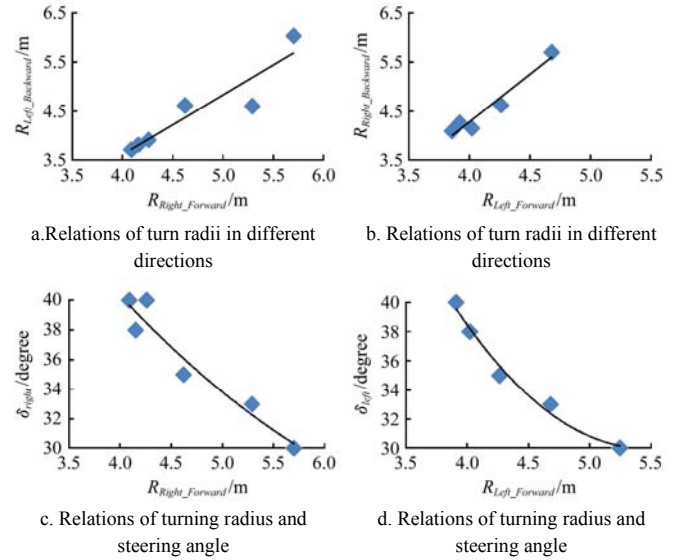


Figure 4 Variance of turning radii of different steering angles

In case of overfitting, a linear function indicates the relation of turning radii in different directions:

$$\begin{cases} R_{Left_Backward} = \alpha_0 \cdot R_{Right_Forward} + \beta_0 \\ R_{Right_Backward} = \alpha_1 \cdot R_{Left_Forward} + \beta_1 \end{cases} \quad (6)$$

where, $R_{Left_Backward}$ and $R_{Left_Forward}$ are the turning radii of a turning left maneuver when the tractor moves backward or forward, respectively; $R_{Right_Backward}$ and $R_{Right_Forward}$ represent the turning radii of backward and forward movements with steering to right, respectively. Steering angle for headland turning is represented as a quadratic function of turning radius in:

$$\begin{cases} \delta_{right} = s_{00} \cdot R_{Right_Forward}^2 + s_{01} \cdot R_{Right_Forward} + s_{02} \\ \delta_{left} = s_{10} \cdot R_{Left_Forward}^2 + s_{11} \cdot R_{Left_Forward} + s_{12} \end{cases} \quad (7)$$

where, δ_{right} and δ_{left} are steering angles on the right and left. Large or maximum steering action is commonly used at headland turns, and experiments in this study reveal that the quadratic function fit the data better than a tangent function in expressing the nonlinear relation of turning radius and steering angle^[19]. Equations (6) and (7) compose the vehicle model for designing and optimizing navigation path. The coefficients of the vehicle model can be rewritten as:

$$r = \begin{bmatrix} \alpha_0 & \beta_0 \\ \alpha_1 & \beta_1 \end{bmatrix} \text{#####} \quad (8)$$

and

$$s = \begin{bmatrix} s_{00} & s_{01} & s_{02} \\ s_{10} & s_{11} & s_{12} \end{bmatrix} \quad (9)$$

During a field test, the vehicle model should be tuned based on the result of each turn. Turning radii of forward and backward movements represent the state of the tractor in the current field. In case raw measured data are not consistent, data from different experiments are processed and combined with the initial dataset. The update of weights in the dataset follows the rules in:

$$\omega_j = \begin{cases} \frac{1}{T-1}, j \in [0, N_0 - 1] \\ N_0 + \sum_{i=0}^j \eta_i \\ \frac{\eta_j}{j - N_0}, j \in [N_0, N - 1] \\ N_0 + \sum_{i=0}^j \eta_i \end{cases}$$

$$\eta = \max(0, 1 - \mu^* \|R_m - R'_m\|) \text{#####} \quad (10)$$

s.t. $\mu \in [0, 1], N \in [N_0, 2N_0]$

where, μ , set as a constant value, is the learning rate of the dynamic turning method. R_m is the predicted turning radius of the vehicle model, and R_m' is the measured value by Equation (4). Processed data for calculating the initial vehicle model are stored from number 0 to N_0-1 continuously and share the same value of weight. The real-time measured data indexed from N_0 to $N-1$ are added to the dataset with different weights, which are related to the errors of the model during each field test. In case of overwhelming data, the weights of experiments are limited to a state smaller than the weights of data in the original dataset, and the amount of dataset (N) is defined to be smaller than a threshold (such as two times of N_0 in this study). The reinforcement learning mechanism tunes the model and adapts it to different conditions, and also guarantees robustness.

3 Experiments and discussion

In order to investigate the performance of the newly developed turning algorithm, several tests were conducted at an experimental farm of Hokkaido University, Sapporo, Hokkaido, Japan. The robotic platforms in this study were two four-wheel-type tractors (EG105 and EG83, YANMAR Co., Ltd., Japan) as shown in Figure 5. There was no implement connected to the EG105^[21], and a rotary was connected to the EG83. An RTK-GPS system (Trimble SPS855, Trimble Navigation, USA) and an inertial measurement unit (IMU) (VN100, VectorNav Technologies, USA) were installed inside the cabin to measure the tractor's absolute position and the posture of the tractor, respectively. A PC in the robot tractor with Windows® operation system (OS) is used to perform all the functions including robot control, obstacle detection and receiving feedback information from the tractor. The control algorithm in this paper is mainly developed in the C++ programming language.

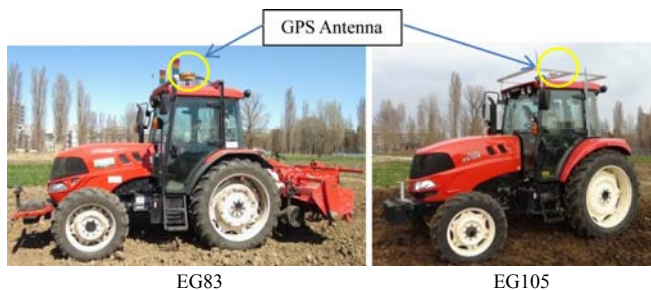


Figure 5 Research platforms

3.1 Initial parameter identification

Figure 4 shows the data for calculating the vehicle model of the tractor, EG105. The turning radius will change with the speed and soil conditions. To simplify the experiment, the position data of the tractor are recorded at a speed of 1.0 m/s in different soil conditions. In the experiments, the tractor steers at $\sigma = 30^\circ, 33^\circ, 35^\circ, 38^\circ$ and 40° to the right and the left as shown in Figures 4c and 4d, separately. Each steering angle is paired with one turning radius or more values (like the 40° in Figure 4c). Therefore, the number of training dataset (N_0) is 11 in this example. The amount of training dataset was defined to be 20, that is to say, the vehicle model would be tuned according to the previous 9 turns. Initial values of the vehicle model are:

$$r = \begin{bmatrix} 1.22 & -1.28 \\ 1.95 & -3.51 \end{bmatrix} \quad \#\# \quad (11)$$

and

$$s = \begin{bmatrix} 0.85 & -14.12 & 83.21 \\ 4.02 & -43.86 & 149.52 \end{bmatrix} \quad \#\#\# \quad (12)$$

3.2 Overall performance evaluation

The trajectory of the robot tractor EG83 is shown in Universal Transverse Mercator (UTM) coordinate system in Figure 6. There are 10 paths from the right to left of the map, and the path space was set to 2.5 m. The reference headland distance was set to 9.5 m. The working order of the paths was in sequence, from Path 1 (near the arrow in Figure 6) to Path 10 with 9 turns. The minimum speed for turning was set as 0.7 m/s. Due to the oscillations caused by the uneven ground surface at the headland, the field test was conducted up to 1.2 m/s turning speed.

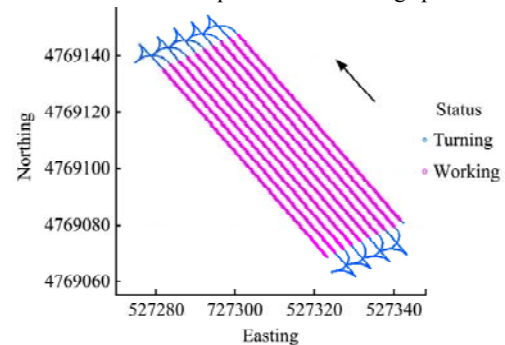


Figure 6 Trajectory of the robot tractor

Field tests were done 6 times with different combinations of weights in the Equation (1). As shown in Table 2, the efficiency and accuracy of the robot tractor were analyzed from 5 aspects: occupied headland distance, length of the turning trajectory, time consumption, average speed during turning, and lateral deviation of the end-point. According to the data shown in Table 2, it can be concluded that the end-point deviation is smaller than 5 cm when the headland distance is larger than 8.02 m.

Table 2 Field tests with different weights combinations

	$[h, t, \varepsilon]$	[0, 1, 0]	[0, 0, 1]	[1, 0, 0]	[0.1, 0.6, 0.3]	[0.6, 0.4, 0]	[0.4, 0.2, 0.4]
Headland/m	9.75	9.26	6.62	9.31	8.02	8.34	
Trajectory/m	28.7	27.4	20.4	27.4	24.2	25.0	
Time/s	30.7	33.8	25.1	32.4	27.1	30.7	
Average speed/m·s ⁻¹	0.93	0.81	0.81	0.85	0.89	0.81	
Deviation/cm	3.9	2.3	13.8	4.1	5.1	2.6	

The DCB turning algorithm can regulate maneuvers of the robot at the headland according to the constraints of the field and manual settings. Taking the third turn of the 10-path map as an example, trajectories of the robot tractor are shown in Figure 7. To express the length of the trajectory clearly, the origin of the coordinate was shifted to (527180.0, 4769333.0) in UTM coordinate. The red line indicates the edge of the field, and the black line is the extension line of the next working path. Firstly, the robot turned along the red circles and went backward along the black dots. Then the robot tractor moved into the next path as indicated by blue blocks. The turning paths are different from each other because of the change of weights in the Equation (1). For example, the robot tractor's trajectory is shown in Figure 7a when the ratio of t is set to 1. The robot steers to a certain steering angle to occupy the whole headland space and turns at the maximum speed. In Figure 7c, the robot fully steers to the left side and turns at a low speed so as to decrease the headland distance because the ratio of h is set to 1. In Table 2, it is observed that the headland distance and the turning trajectory in the third column are smaller than values in other conditions. However, the average lateral deviation after turning is 13.8 cm, which is much larger than deviations in other

conditions. Even though the robot tractor might deviate from the next path after the backward movement as shown in Figures 7a and 7c, the robot tractor can make use of the headland space and approach the next path smoothly. The weights should be

adjusted according to the requirements of headland distance, time, and end-point accuracy. Accordingly, the robot tractor will turn to next working path along different trajectories similar to those in Figures 7d-7f.

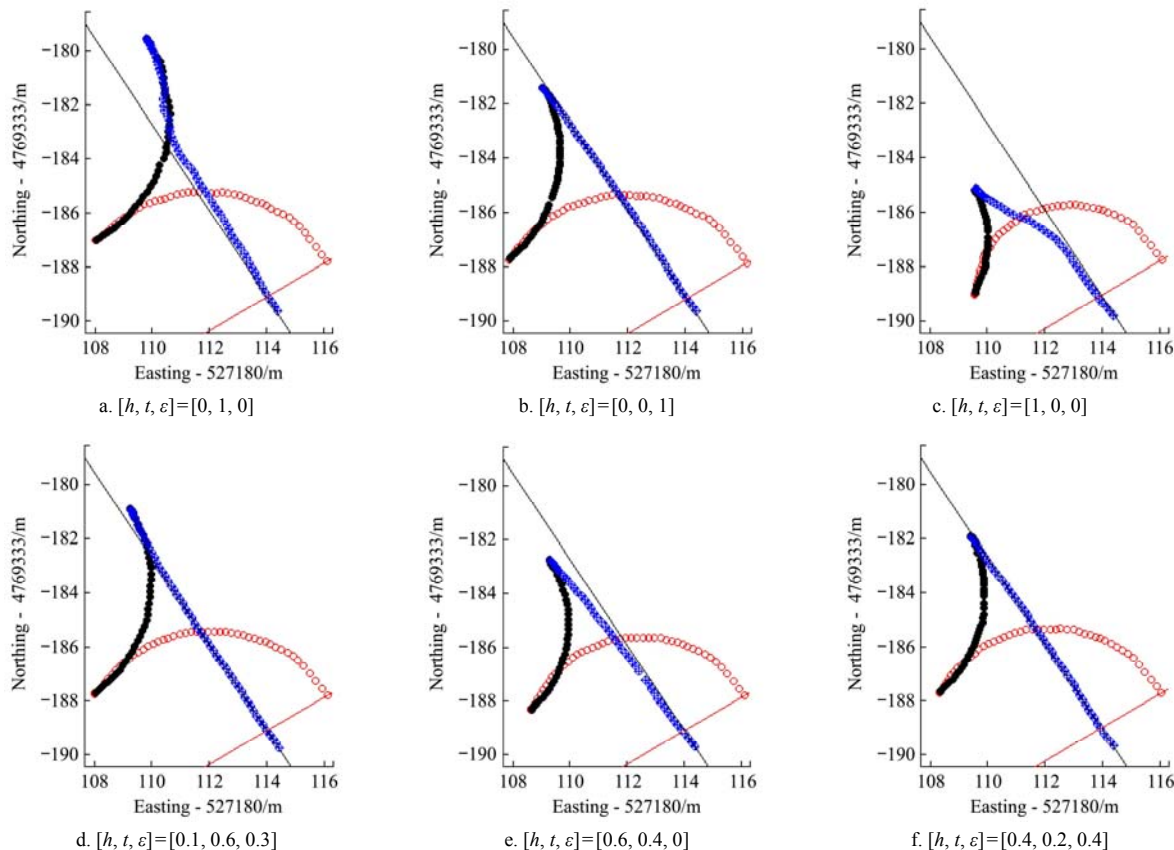
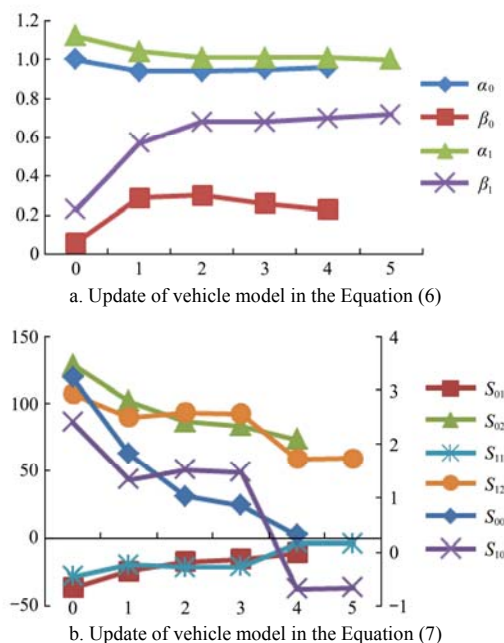


Figure 7 Trajectories of the robot tractor during the third turn

Updating the vehicle model during the field test is a key factor for the robot tractor to follow the path accurately. When the weight of accuracy (ϵ) was set to 1, the robot tractor should ideally stop at the extension line of the next path (point C_2 in Figure 3) after the backward movement as shown in Figure 7b. However, deviation from the planned path is inevitable because of various factors such as the error of the initial vehicle model, sideslip of the robot tractor when turning, the time delay of the control unit. The procedure of updating the vehicle model in DCB turning algorithm is shown in Figure 8. For the first turning, the vehicle steers to right/left with the initial model. It can be observed that the vehicle model became stable after tuning parameters one time. In addition, gaps of parameters after the 3rd update illustrate that the model is adaptive to the variations of external factors and converges again soon. The trajectory of the first turn is shown in Figure 9a. For comparison, the trajectory of the third turn in Figure 7b is enlarged and is shown in Figure 9b. It is found that the trajectory of the robot tractor coincided with the navigation path after one-time parameter tuning.

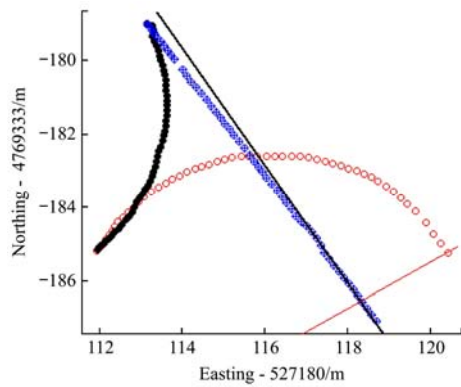
Figure 10 shows changes in vehicle speed and steering angle during a turn. The speed during a turn was 1.0 m/s and the speed in operation was about 1.4 m/s. The maximum steering angle was ± 40 degree during the turn. The graphs also show that the robot tractor did not stop when the steering angle was changed. Although the robot tractor deviates from the optimal path when the steering angle is changed, the method for estimating the turning radius in real time can decrease the error caused by deviation.

Moreover, the reinforcement learning method is used to update the vehicle model and to regulate the turning angle of the basis of the experimental data. Therefore, the robot tractor could turn into the next path with the precision of 3.9 cm at a speed of 1.2 m/s when the ratio of t was 1.

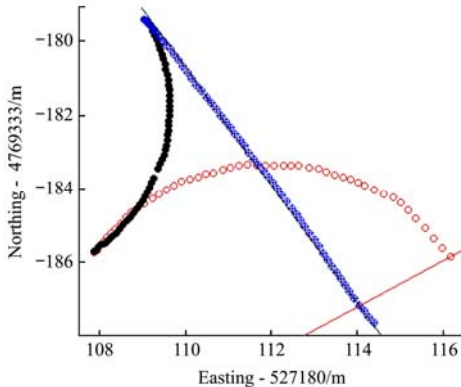


Note: S_{00} and S_{10} are plotted on the secondary axis.

Figure 8 Update of vehicle model parameters



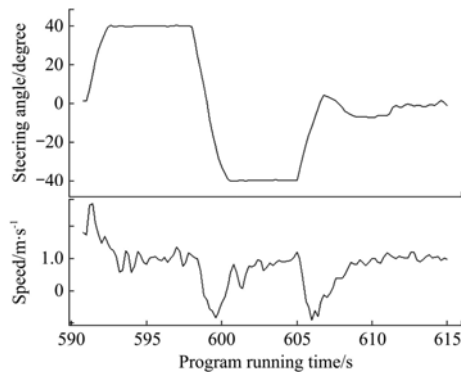
a. Trajectory of the first turn



b. Trajectory of the third turn

Note: Weights of the objective function were set to $[h, t, \varepsilon]=[0, 0, 1]$.

Figure 9 Trajectories of the robot tractor when turning



Note: Minus value of steering angle means the tractor steers to left side.

Figure 10 Steering angle and speed when turning

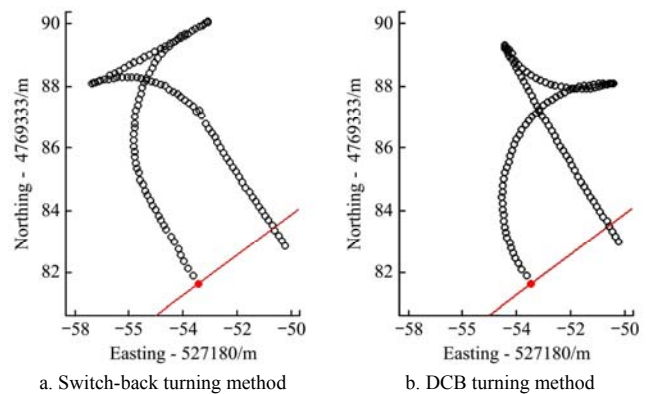
It is proved that the steering angle during a turn could be adjusted on the basis of the desired headland distance. When the headland space is large enough for turning, the robot tractor turns to next path with small steering angle, which is important for not damaging the soil surface. Besides, by calculating the radius of the trajectory during a turn, the algorithm could optimize the planned path in real time. With the learning mechanism, the DCB algorithm could analyze the accuracy of measured data after each turn, tune the vehicle model and make it applicable to different conditions, and guarantee robustness. One limitation of the DCB

turning control algorithm is that it is not suitable for the tractor with a towed implement system, because the position of the implement (i.e., a trailer) is controlled by several actuators including the steering of the tractor and the hydraulically controlled joint. But this turning algorithm works well for the tractor with a mounted implement, such as the rotary in Figure 5, the plow and the sprayer, which does not move horizontally relative to the tractor.

3.3 Comparison to the switch-back turning algorithm

The switch-back turning method^[8] is now widely used because of its good efficiency and accuracy. Therefore, the switch-back turning as a representation of traditional methods was compared with the DCB turning method in four maps with different path space, which are 2.0 m, 2.5 m, 3.0 m and 3.5 m. The robot tractor in this experiment was EG105 without an implement behind the tractor. The speed of turning for the two methods was the same, 1.0 m/s. For DCB turning, the weights were set to $[0.5, 0, 0.5]$ in the tests, which meant h and ε were equally considered. Field experiments were conducted in maps with 10 working paths for both the switch-back turning and DCB turning methods. The reference headland distance was set to 7.0 m, and the tractor therefore fully steered to the left/right at the end of the path. For the switch-back turning method, the forward movement before turning was 2.0 m so as to approach to the next lane with a small deviation. Note: Path space was 3.5 m.

Figure 11 shows trajectories of the robot tractor with the different turning algorithms. The robot tractor started to turn at the red point in Figure 11 and steered fully to the right in this example.



a. Switch-back turning method

b. DCB turning method

Note: Path space was 3.5 m.

Figure 11 Turning trajectories of the robot tractor

As shown in Table 3, four aspects were analyzed in the comparison experiments: length of turning trajectory, time consumption, occupied headland distance, and lateral deviation of the end-point. The maximum headland occupancy of DCB turning was not larger than that of switch-back turning. The trajectory of DCB turning was 21% less than that of switch-back turning when the width of two paths was 3.0 m. Corresponding to the decrease in the trajectory, the time consumption of DCB turning was 17% less than that of the switch-back turning algorithm. When the DCB turning algorithm was used, the improvement of accuracy after turning was about 79% when the distance between paths was set to 2.0 m.

Table 3 Comparison of DCB turning and the switch-back turning algorithm

W/m	2.0		2.5		3.0		3.5	
	DCB	Switch-back	DCB	Switch-back	DCB	Switch-back	DCB	Switch-back
Trajectory/m	21.7	25.5	20.9	23.9	20.0	25.3	19.8	24.2
Time/s	26	31	26	30	25	30	25	30
Headland/m	7.4	7.4	6.9	7.0	6.5	7.3	6.2	7.4
Deviation/cm	3.6	17.3	5.4	14.2	5.6	16.4	7.7	19.2

4 Conclusions

An adaptive headland turning algorithm, DCB, was presented in this paper. The algorithm could create an efficient path, with connected primitives, and optimize maneuvers during a turn. The vehicle model in DCB turning was represented as the relation of turning radii in different directions and the relation of turning radius and steering angle in forward movement. The trajectory pattern and the dynamic vehicle model in the DCB turning algorithm make it more accurate than the traditional switch-back turning method in the same headland distance. Under the control of DCB turning algorithm, the experimental robot tractor could turn to the next working path with or without a rotary at a speed of 1.2 m/s. The average lateral deviation after turning was about 3.9 cm, which was a significant improvement compared with the switch-back turning method. And it brings less headland occupation and time consumption as well. Therefore, it can be concluded that the adaptive turning algorithm is accurate and robust enough for controlling robot tractors.

Acknowledgements

This study was supported by the R&D program of SIP (Cross-Ministerial Strategic Innovation Promotion Program) “Technologies for Creating Next-generation Agriculture, Forestry and Fisheries”.

Nomenclature

a, b	center position of the fitted circle (turning trajectory)
h	weight of headland distance
H_0	reference headland distance
N_0	number of initial dataset
N	number of whole dataset
r	coefficient matrix of vehicle model indicating the relation of turning radii in different directions
R	turning radius (Subscripts indicate the moving direction.)
s	coefficients/matrix of vehicle model indicating the relation of steering angle and turning radii
t	weight of efficiency
j	index of dataset
J	objective function
v	vehicle speed
V	limitation of speed
W	distance between the current path and the next path
(x, y)	2D position of the robot tractor recorded by the RTK-GPS
α, β	coefficients of vehicle model indicating the relation of turning radii in different directions
σ	steering angle
Δd	lateral deviation
ε	weight of accuracy
ω	weight of each datum in the dataset
μ	learning rate of the dynamic turning method
θ	change of yaw angle

[References]

- [1] Torisu R, Tanaka K, Imae J, Ishikawa T. Optimal Path of Headland for Tractors by Optimal Control Theory (Part1): Formulation of Optimal Control Problems and Forward Maneuver of Tractors. *J Japanese Soc Agric Mach*, 1997; 59: 3–10.
- [2] Takai R, Barawid O, Noguchi N. Autonomous navigation system of crawler-type robot tractor. *IFAC Proc*, 2011; 18: 14165–14169.
- [3] Torisu R, Tanaka K, Imae J. Optimal path of headland for tractors by optimal control theory (Part 2): Combination of forward and backward maneuvers. *J Japanese Soc Agric Mach*, 1998; 60: 45–53.
- [4] Kise M, Noguchi N, Ishii K, Terao H. Field mobile robot navigated by RTK-GPS and FOG (Part 2): Autonomous operation by applying navigation map. *J Japanese Soc Agric Mach*, 2001; 63: 80–85.
- [5] Jin J. Optimal field coverage path planning on 2D and 3D surfaces. Iowa State University, 2009.
- [6] Tu X. Robust navigation control and headland turning optimization of agricultural vehicles. Iowa State University, 2013.
- [7] Yang L. Development of a robot tractor implemented an omni-directional safety system. Hokkaido University, 2013.
- [8] Kise M, Noguchi N, Ishii K, Terao H. Field mobile robot navigated by RTK-GPS and FOG (Part 3): Enhancement of turning accuracy by creating path applied with motion constraints. *J Japanese Soc Agric Mach*, 2002; 64: 102–110.
- [9] Cariou C, Lenain R, Thuilot B, Humbert T, Berducot M. Maneuvers automation for agricultural vehicle in headland. *AgEng 2010 Conf*, Clermont-Ferrand, Fr 2010: 1–10.
- [10] Sabelhaus D, Lammers P S, Peter L, Röben F. Path planning of headland turn manoeuvres. *Landtechnik*, 2015; 70(4): 123–131.
- [11] Backman J, Piirainen P, Oksanen T. Smooth turning path generation for agricultural vehicles in headlands. *Biosyst Eng*, 2015; 139: 76–86.
- [12] Kise M, Noguchi N, Ishii K, Terao H. Field mobile robot navigated by RTK-GPS and FOG (Part 4): The steering controller applied optimal controller. *J Japanese Soc Agric Mach*, 2002; 64: 76–84.
- [13] Kraus T, Ferreau H J, Kayacan E, Ramon H, de Baerdemaeker J, Diehl M, et al. Moving horizon estimation and nonlinear model predictive control for autonomous agricultural vehicles. *Comput Electron Agric*, 2013; 98: 25–33.
- [14] Karkee M, Steward B L. Study of the open and closed loop characteristics of a tractor and a single axle towed implement system. *J Terramechanics*, 2010; 47: 379–393.
- [15] Backman J, Oksanen T, Visala A. Navigation system for agricultural machines: Nonlinear model predictive path tracking. *Comput Electron Agric*, 2012; 82: 32–43.
- [16] Yoshida Y, Wang Q, Oya M, Okumura K. Adaptive longitudinal velocity and lane keeping control of four-wheel-steering vehicles. *2007 SICE Annu Conf*, IEEE, 2007; pp. 1305–1310.
- [17] Oya M, Wang Q. Adaptive lane keeping controller for four-wheel-steering vehicles. *2007 IEEE Int. Conf. Control Autom.*, IEEE, 2007; pp. 1942–1947.
- [18] Petrov P, Nashashibi F. Modeling and nonlinear adaptive control for autonomous vehicle overtaking. *IEEE Trans Intell Transp Syst*, 2014; 15: 1643–1656.
- [19] Tashiro T. Vehicle steering control with MPC for target trajectory tracking of autonomous reverse parking. *2013 IEEE Int. Conf. Control Appl.*, IEEE, 2013; pp. 247–251.
- [20] Wang H, Noguchi N. Autonomous maneuvers of a robotic tractor for farming. *2016 IEEE/SICE Int. Symp. Syst. Integr.*, IEEE, 2016; pp. 592–597.
- [21] YANMAR. Main Specifications of Tractor EG97 • EG105. <https://www.yanmar.com/jp/agri/products/tractor/eg97-105/spec/> [accessed on November 10, 2017].